Debugging in Java using IntelliJ Q&A:

What is the purpose of a breakpoint?

- A breakpoint is a debugging feature used by developers to pause the execution of a program or script at a specific line of code.
- It allows developers to inspect the program's state, variables, and memory at that particular point in the code.
- By setting breakpoints strategically, developers can analyse and identify the cause of bugs or unexpected behaviour in their code more effectively.

Does the line of code on a breakpoint run when you start debugging?

- No, when you start debugging, the line of code where a breakpoint is set will not execute immediately.
- Instead, the debugger will pause the execution of the program just before it reaches the breakpoint.
- This gives developers an opportunity to examine the program's state and make any necessary adjustments or observations before continuing the execution.

How do we debug the next line of code?

- To debug the next line of code after hitting a breakpoint, you can use the "Step Over" command, (usually available in debugging tools and integrated development environments).
- When you choose "Step Over," the debugger will execute the current line of code and then pause again at the next line of code.
- This allows you to proceed with the debugging process one line at a time.

What does the step into command do?

- The "Step Into" command is another debugging feature that allows developers to delve deeper into the code by stepping into a function or method call.
- If the current line of code contains a function or method call, using "Step Into" will make the debugger pause at the first line of that function or method.
- This allows you to examine the internal workings of the function or method and debug it if necessary.

What is the difference between evaluate expression and evaluate code fragment?

- In most debugging environments, **"Evaluate Expression"** and **"Evaluate Code Fragment"** are features that enable developers to inspect and manipulate variables and expressions during debugging.

The key difference between the two lies in their scope and complexity:

- **Evaluate Expression:** This feature allows developers to evaluate and view the result of a single expression or variable at a specific point in the code.
- It is designed to work with simple expressions and typically does not support more complex statements or multiple lines of code.
- **Evaluate Code Fragment:** This feature allows developers to execute a small block of code (multiple lines) and view the result.
- It is more versatile than "Evaluate Expression" as it can handle more complex statements and even run multiple lines of code to observe their effects.

To summarise, **"Evaluate Expression"** is for simple, single expressions or variables, while **"Evaluate Code Fragment"** is for more complex code snippets or multiple lines of code.